Организация адресного пространства памяти

Материалы по дисциплине «Микропроцессорные системы»
Специальность «Компьютерные системы и комплексы»
Составитель: Торгашин Р.Г

ГБПОУ ВО "Борисоглебский техникум промышленных и информационных технологий"

2016 год

Оглавление

Физическая организация памяти компьютера

Запоминающие устройства компьютера разделяют, как минимум, на два уровня: основную (главную, оперативную, физическую) и вторичную (внешнюю) память.

Основная память представляет собой упорядоченный массив однобайтовых ячеек, каждая из которых имеет свой уникальный адрес (номер). Процессор извлекает команду из основной памяти, декодирует и выполняет ее. Для выполнения команды могут потребоваться обращения еще к нескольким ячейкам основной памяти. Обычно основная память изготавливается с применением полупроводниковых технологий и теряет свое содержимое при отключении питания.

Вторичную память (это главным образом диски) также можно рассматривать как одномерное линейное адресное пространство, состоящее из последовательности байтов. В отличие от оперативной памяти, она является энергонезависимой, имеет существенно большую емкость и используется в качестве расширения основной памяти.

Эту схему можно дополнить еще несколькими промежуточными уровнями, как показано на рисунке. Разновидности памяти могут быть объединены в иерархию по убыванию времени доступа, возрастанию цены и увеличению емкости.

Рисунок 1: Иерархия памяти

Многоуровневую схему используют следующим образом. Информация, которая находится в памяти верхнего уровня, обычно хранится также на уровнях с большими номерами. Если процессор не обнаруживает нужную информацию на i-м уровне, он начинает искать ее на следующих уровнях. Когда нужная информация найдена, она переносится в более быстрые уровни.

Оказывается, при таком способе организации по мере снижения скорости доступа к уровню памяти снижается также и частота обращений к нему.

Ключевую роль здесь играет свойство реальных программ, в течение ограниченного отрезка времени способных работать с небольшим набором адресов памяти. Это эмпирически наблюдаемое свойство известно как **принцип локальности** или локализации обращений.

В случае ОС свойство локальности объяснимо, если учесть, как пишутся программы и как хранятся данные, то есть обычно в течение какого-то отрезка времени ограниченный фрагмент кода работает с ограниченным набором данных. Эту часть кода и данных удается разместить в памяти с быстрым доступом. В результате реальное время доступа к памяти определяется временем доступа к

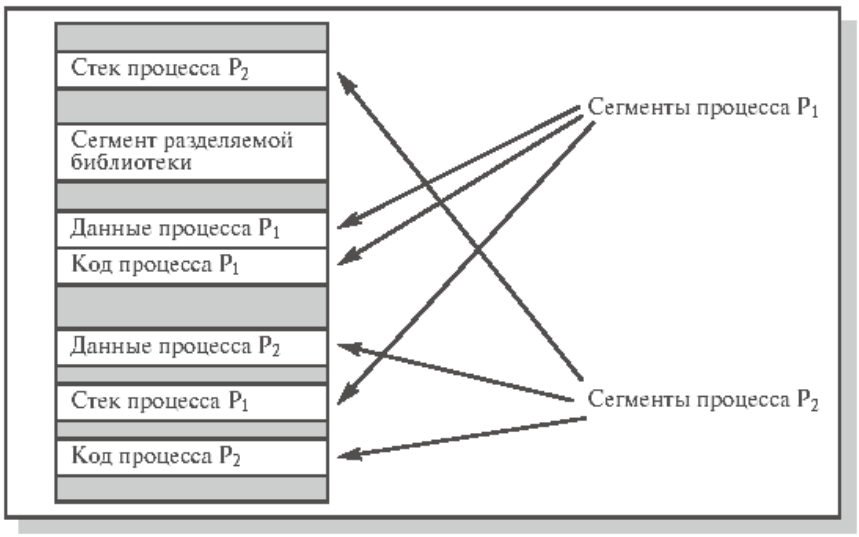

Рисунок 2: Расположение сегментов процессов в памяти компьютера

верхним уровням, что и обусловливает эффективность использования иерархической схемы.

Адреса в основной памяти, характеризующие реальное расположение данных в физической памяти, называются физическими адресами. Набор физических адресов, с которым работает программа, называют физическим адресным пространством.

Логическая память

Таким образом архитектура компьютера различает физическое адресное пространство (ФАП) и логическое адресное пространство (ЛАП).

ФАП является простым массивом байт. Доступ к данным происходит по адресу на шине микропроцессорной системы.

ЛАП организуется программистом.

Преобразование (трансляцию) логических адресов в физические осуществляет блок управления памятью MMU.

В современных системах ЛАП может быть представлен в виде набора структур данных: байт, сегментов, страниц. :

Плоское(линейное) ЛАП

Линейная логическая адресация памяти - массив байт без определенной структуры. Логический адрес совпадает с физическим, поэтому трансляция не нужна. Этот вариант используется только для случаев, когда содержимое памяти расположено по конкретным адресам, известным заранее. Например для таблицы векторов прерываний, которая всегда расположена в ячейках 00h...3FFh. В большинстве остальных случаев физические адреса программы определяются на стадии запуска или выполнения программы.

Аппаратная организация памяти в виде линейного набора ячеек не соответствует представлениям программиста о том, как организовано хранение программ и данных. Большинство программ представляет собой набор модулей, созданных независимо друг от друга. Иногда все модули, входящие в состав процесса, располагаются в памяти один за другим, образуя линейное пространство адресов. Однако чаще модули помещаются в разные области памяти и используются по-разному.

Схема управления памятью, поддерживающая этот взгляд пользователя на то, как хранятся программы и данные, называется сегментацией. **Сегмент** – область памяти определенного назначения, внутри которой поддерживается линейная адресация. Сегменты содержат процедуры, массивы, стек или скалярные величины, но обычно не содержат информацию смешанного типа.

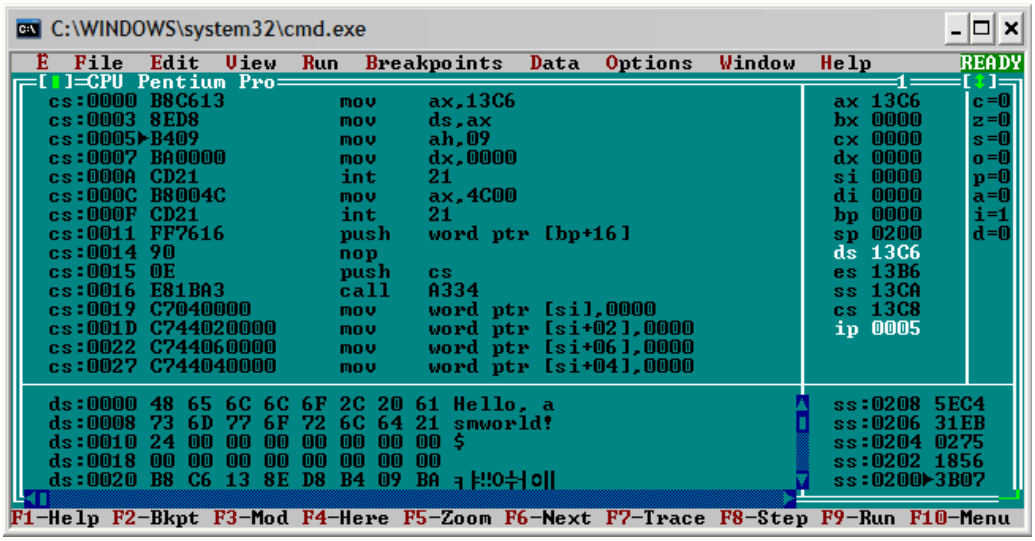

Рисунок 3: Работа программы с сегментами

Большинство современных ОС поддерживают сегментную организацию памяти. В некоторых архитектурах (Intel, например) сегментация поддерживается оборудованием.

Адреса, к которым обращается процесс, таким образом, отличаются от адресов, реально существующих в оперативной памяти. В каждом конкретном случае используемые программой адреса могут быть представлены различными способами. Например, адреса в исходных текстах обычно символические. Компилятор связывает эти символические адреса с перемещаемыми адресами (такими, как n байт от начала модуля). Подобный адрес, сгенерированный программой, обычно называют логическим (в системах с виртуальной памятью он часто называется виртуальным) адресом. Совокупность всех логических адресов называется логическим (виртуальным) адресным пространством.

Связывание адресов

Итак логические и физические адресные пространства ни по организации, ни по размеру не соответствуют друг другу. Максимальный размер логического адресного пространства обычно определяется разрядностью процессора (например, 2^{32}) и в современных системах значительно превышает размер физического адресного пространства. Следовательно, процессор и ОС должны быть способны отобразить ссылки в коде программы в реальные физические адреса, соответствующие текущему расположению программы в основной памяти. Такое отображение адресов называют трансляцией (привязкой) адреса или связыванием адресов.

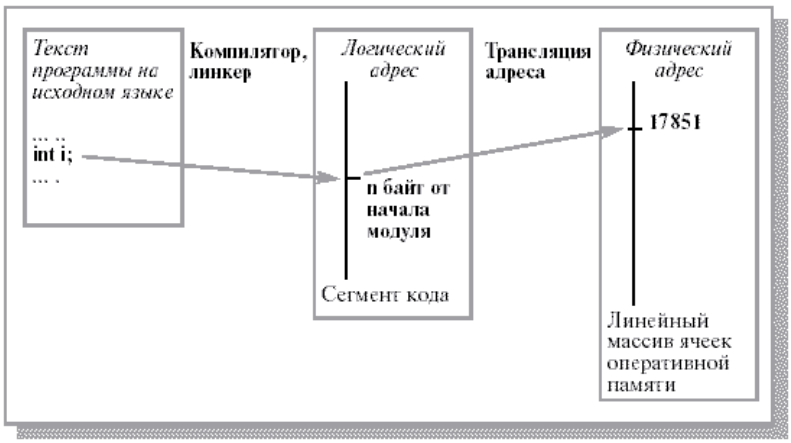

Рисунок 4: Формирование логического адреса и связывание логического адреса с физическим

Связывание логического адреса, порожденного оператором программы, с физическим должно быть осуществлено до начала выполнения оператора или в момент его выполнения. Таким образом, привязка инструкций и данных к памяти в принципе может быть сделана на следующих шагах:

- **Этап компиляции** (Compile time). Когда на стадии компиляции известно точное место размещения процесса в памяти, тогда непосредственно генерируются физические адреса. При изменении стартового адреса программы необходимо перекомпилировать ее код. В качестве примера можно привести .com программы MS-DOS, которые связывают ее с физическими адресами на стадии компиляции.
- **Этап загрузки** (Load time). Если информация о размещении программы на стадии компиляции отсутствует, компилятор генерирует перемещаемый код. В этом случае окончательное связывание откладывается до момента загрузки. Если стартовый адрес меняется, нужно всего лишь перезагрузить код с учетом измененной величины.
- **Этап выполнения** (Execution time). Если процесс может быть перемещен во время выполнения из одной области памяти в другую, связывание откладывается до стадии выполнения. Здесь желательно наличие специализированного оборудования, например регистров перемещения. Их значение прибавляется к каждому адресу, сгенерированному процессом. Большинство современных ОС осуществляет трансляцию адресов на этапе выполнения, используя для этого специальный аппаратный механизм.

Функции системы управления памятью

Чтобы обеспечить эффективный контроль использования памяти, ОС должна выполнять следующие функции:

- отображение адресного пространства процесса на конкретные области физической памяти;
- распределение памяти между конкурирующими процессами;
- контроль доступа к адресным пространствам процессов;
- выгрузка процессов (целиком или частично) во внешнюю память, когда в оперативной памяти недостаточно места;
- учет свободной и занятой памяти.

Простейшие схемы управления памятью

Первые ОС применяли очень простые методы управления памятью. Вначале каждый процесс пользователя должен был полностью поместиться в основной памяти, занимать непрерывную область памяти, а система принимала к обслуживанию дополнительные пользовательские процессы до тех пор, пока все они одновременно помещались в основной памяти. Затем появился "**простой свопинг**" (система по-прежнему размещает каждый процесс в основной памяти целиком, но иногда на основании некоторого критерия целиком сбрасывает образ некоторого процесса из основной памяти во внешнюю и заменяет его в основной памяти образом другого процесса). Такого рода схемы имеют не только историческую ценность. В настоящее время они применяются в учебных и научно-исследовательских модельных ОС, а также в ОС для встроенных (embedded) компьютеров.

Схема с фиксированными разделами

Самым простым способом управления оперативной памятью является ее предварительное (обычно на этапе генерации или в момент загрузки системы) разбиение на несколько разделов фиксированной величины. Поступающие процессы помещаются в тот или иной раздел. При этом происходит условное разбиение физического адресного пространства. Связывание логических и физических адресов процесса происходит на этапе его загрузки в конкретный раздел, иногда – на этапе компиляции.

Каждый раздел может иметь свою очередь процессов, а может существовать и глобальная очередь для всех разделов.

Очевидный недостаток этой схемы – число одновременно выполняемых процессов ограничено числом разделов.

Другим существенным недостатком является то, что предлагаемая схема сильно страдает от внутренней фрагментации – потери части памяти, выделенной процессу, но не используемой им. Фрагментация возникает потому, что процесс не полностью занимает выделенный ему раздел или потому, что некоторые разделы слишком малы для выполняемых пользовательских программ.

Один процесс в памяти

Частный случай схемы с фиксированными разделами – работа менеджера памяти однозадачной ОС. В памяти размещается один пользовательский процесс. Остается определить, где располагается пользовательская программа по отношению к ОС – в верхней части памяти, в нижней или в средней. Причем часть ОС может быть в ROM (например, BIOS, драйверы устройств). Главный фактор, влияющий на это решение, – расположение вектора прерываний, который обычно локализован в нижней части памяти, поэтому ОС также размещают в нижней. Примером такой организации может служить ОС MS-DOS.

Защита адресного пространства ОС от пользовательской программы может быть организована при помощи одного граничного регистра, содержащего адрес границы ОС.

Оверлейная структура

Так как размер логического адресного пространства процесса может быть больше, чем размер выделенного ему раздела (или больше, чем размер самого большого раздела), иногда используется техника, называемая оверлей (overlay) или организация структуры с перекрытием. Основная идея – держать в памяти только те инструкции программы, которые нужны в данный момент.

На современных 32-разрядных системах, где виртуальное адресное пространство измеряется гигабайтами, проблемы с нехваткой памяти решаются другими способами

Динамическое распределение. Свопинг

Имея дело с пакетными системами, можно обходиться фиксированными разделами и не использовать ничего более сложного. В системах с разделением времени возможна ситуация, когда память не в состоянии содержать все пользовательские процессы. Приходится прибегать к свопингу (swapping) – перемещению процессов из главной памяти на диск и обратно целиком. Частичная выгрузка процессов на диск осуществляется в системах со страничной организацией (paging) и будет рассмотрена ниже.

Страничная память

Описанные выше схемы недостаточно эффективно используют память, поэтому в современных схемах управления памятью не принято размещать процесс в оперативной памяти одним непрерывным блоком.

В самом простом и наиболее распространенном случае страничной организации памяти (или paging) как логическое адресное пространство, так и физическое представляются состоящими из наборов блоков или страниц одинакового размера. При этом образуются логические страницы (page), а соответствующие единицы в физической памяти называют физическими страницами или страничными кадрами (page frames). Страницы (и страничные кадры) имеют фиксированную длину, обычно являющуюся степенью числа 2, и не могут перекрываться. Каждый кадр содержит одну страницу данных. При такой организации внешняя фрагментация отсутствует, а потери из-за внутренней фрагментации, поскольку процесс занимает целое число страниц, ограничены частью последней страницы процесса.

Логический адрес в страничной системе – упорядоченная пара (p,d), где p – номер страницы в виртуальной памяти, а d – смещение в рамках страницы p, на которой размещается адресуемый элемент. Заметим, что разбиение адресного пространства на страницы осуществляется вычислительной системой незаметно для программиста. Поэтому адрес является двумерным лишь с точки зрения операционной системы, а с точки зрения программиста адресное пространство процесса остается линейным.

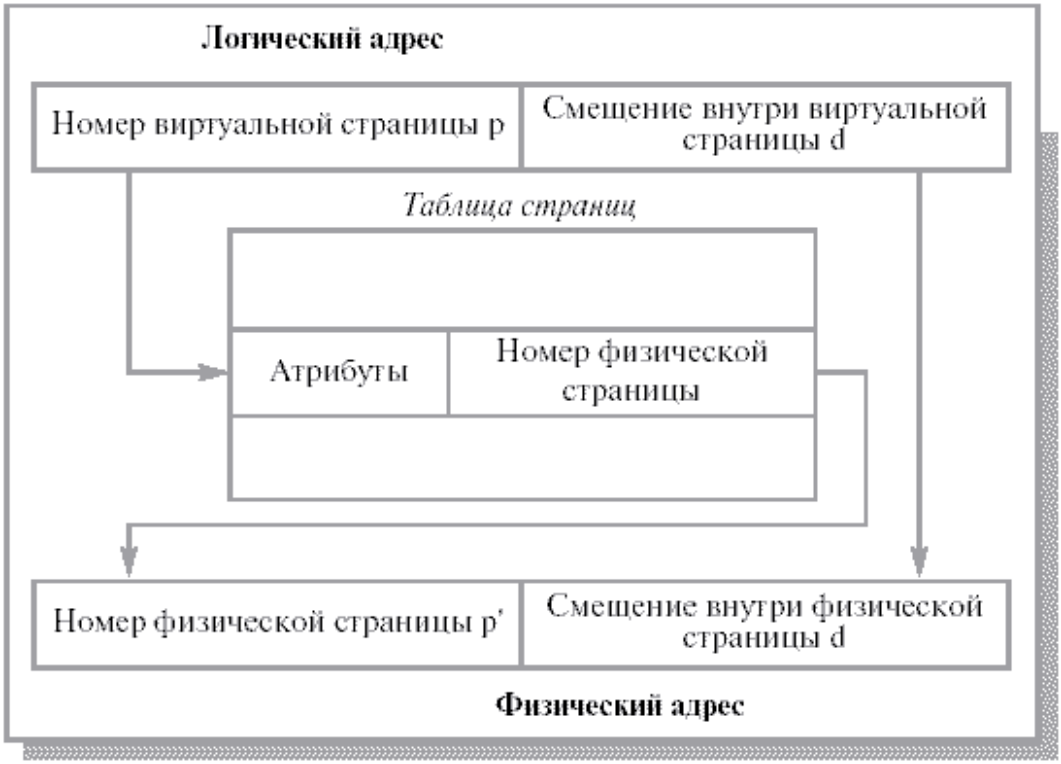

Рисунок 5: Связь логического и физического адресов при страничной организации памяти

Таблица страниц (page table) адресуется при помощи специального регистра процессора и позволяет определить номер кадра по логическому адресу. Помимо этой основной задачи, при помощи атрибутов, записанных в строке таблицы страниц, можно организовать контроль доступа к конкретной странице и ее защиту.

Отметим еще раз различие точек зрения пользователя и системы на используемую память. С точки зрения пользователя, его память – единое непрерывное пространство, содержащее только одну программу. Реальное отображение скрыто от пользователя и контролируется ОС. Заметим, что процессу пользователя чужая память недоступна. Он не имеет возможности адресовать память за пределами своей таблицы страниц, которая включает только его собственные страницы.

Для управления физической памятью ОС поддерживает структуру таблицы кадров. Она имеет одну запись на каждый физический кадр, показывающий его состояние.

Отображение адресов должно быть осуществлено корректно даже в сложных случаях и обычно реализуется аппаратно. Для ссылки на таблицу процессов используется специальный регистр. При переключении процессов необходимо найти таблицу страниц нового процесса, указатель на которую входит в контекст процесса.

Сегментная и сегментно-страничная организация памяти

Существуют две другие схемы организации управления памятью: сегментная и сегментно-страничная. Сегменты, в отличие от страниц, могут иметь переменный размер. Идея сегментации изложена во введении. При сегментной организации виртуальный адрес является двумерным как для программиста, так и для операционной системы, и состоит из двух полей – номера сегмента и смещения внутри сегмента. Подчеркнем, **что в отличие от страничной организации, где линейный адрес преобразован в двумерный операционной системой для**

удобства отображения, здесь двумерность адреса является следствием представления пользователя о процессе не в виде линейного массива байтов, а как набор сегментов переменного размера (данные, код, стек...).

Программисты, пишущие на языках низкого уровня, должны иметь представление о сегментной организации, явным образом меняя значения сегментных регистров (это хорошо видно по текстам программ, написанных на Ассемблере). Логическое адресное пространство – набор сегментов. Каждый сегмент имеет имя, размер и другие параметры (уровень привилегий, разрешенные виды обращений, флаги присутствия). В отличие от страничной схемы, где пользователь задает только один адрес, который разбивается на номер страницы и смещение прозрачным для программиста образом, в сегментной схеме пользователь специфицирует каждый адрес двумя величинами: именем сегмента и смещением.

В системах, где сегменты поддерживаются аппаратно, параметры сегментов обычно хранятся в таблице дескрипторов сегментов, а программа обращается к этим дескрипторам по номерам-селекторам. При этом в контекст каждого процесса входит набор сегментных регистров, содержащих селекторы текущих сегментов кода, стека, данных и т. д. и определяющих, какие сегменты будут использоваться при разных видах обращений к памяти. Это позволяет процессору уже на аппаратном уровне определять допустимость обращений к памяти, упрощая реализацию защиты информации от повреждения и несанкционированного доступа.

Аппаратная поддержка сегментов распространена мало (главным образом на процессорах Intel). В большинстве ОС сегментация реализуется на уровне, не зависящем от аппаратуры.

Хранить в памяти сегменты большого размера целиком так же неудобно, как и хранить процесс непрерывным блоком. Напрашивается идея разбиения сегментов на страницы. При сегментно-страничной организации памяти происходит двухуровневая трансляция виртуального адреса в физический. В этом случае логический адрес состоит из трех полей: номера сегмента логической памяти, номера страницы внутри сегмента и смещения внутри страницы. Соответственно, используются две таблицы отображения – таблица сегментов, связывающая номер сегмента с таблицей страниц, и отдельная таблица страниц для каждого сегмента.

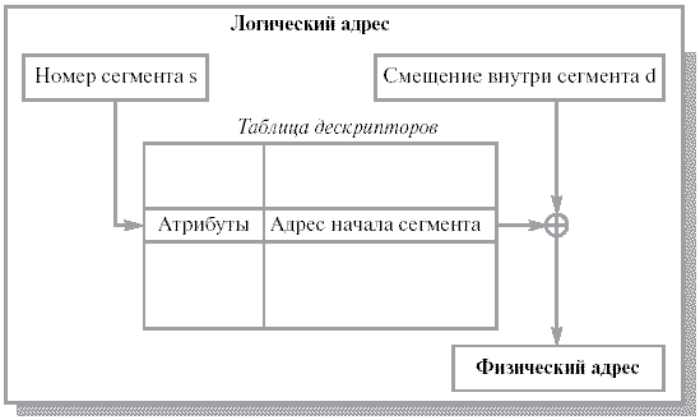

Рисунок 6: Преобразование логического адреса при сегментной организации памяти

Сегментно-страничная и сегментная организация памяти позволяет легко организовать совместное использование одних и тех же данных и программного кода разными задачами. Для этого различные логические блоки памяти разных процессов отображают в один и тот же блок физической памяти, где размещается разделяемый фрагмент кода или данных.

Рисунок 7: Упрощенная схема формирования физического адреса при сегментно-страничной организации памяти

Отображение памяти

Для упрощения работы с программ устройствами большинство PCI устройств отображают свои управляющие регистры на адреса памяти и высокопроизводительные приложения предпочитают иметь прямой доступ к регистрам, вместо того, чтобы постоянно вызывать ioctl() для выполнения этой работы.

Отображение устройства означает связывание диапазона адресов пользовательского пространства с памятью устройства. Всякий раз, когда программа читает или записывает в заданном диапазоне адресов, она на самом деле обращается к устройству.

Существенным ограничением отображения памяти (mmap) является то, что ядро может управлять виртуальными адресами только на уровне таблиц страниц, таким образом, отображённая область должна быть кратной размеру страницы RAM (PAGE_SIZE) и должна находиться в физической памяти начиная с адреса, который кратен PAGE_SIZE.

Распределение памяти

Всё адресное пространство памяти современной системы на базе архитектуры IA-32 можно разделить на следующие крупные участки:

- унаследованный диапазон адресов (Legacy Address Range);
- нижний диапазон адресов главной памяти (Main Memory Address Range);
- нижний диапазон адресов памяти PCI (PCI Memory Address Range);
- верхний диапазон адресов главной памяти (Main Memory Address Range);
- диапазон адресов перемещённой главной памяти (Main Memory Reclaim Address Range);
- верхний диапазон адресов памяти PCI (PCI Memory Address Range).

Унаследованный диапазон адресов

Младший мегабайт адресного пространства (физические адреса от 0000_0000 до 000F_FFFF включительно) называется **унаследованным диапазоном адресов** (Legacy Address Range), поскольку его структура определяется требованиями совместимости с ранними моделями ПК. Он состоит из нескольких частей.

Нижние 640 Кбайт (адреса 0000_0000 — 0009_FFFF) всегда отображаются на оперативную память: это так называемая **стандартная память**.

Далее следуют 128 Кбайт (адреса 000A_0000 — 000B_FFFF), которые на ранних ПК использовались **для доступа к видеопамяти**. Конкретная их структура зависела от типа используемого видеоадаптера и его режима работы; например, цветные графические адаптеры в графических режимах использовали видеопамять, начинающуюся с адреса A0000, а информация в текстовых режимах размещалась с адреса B8000.

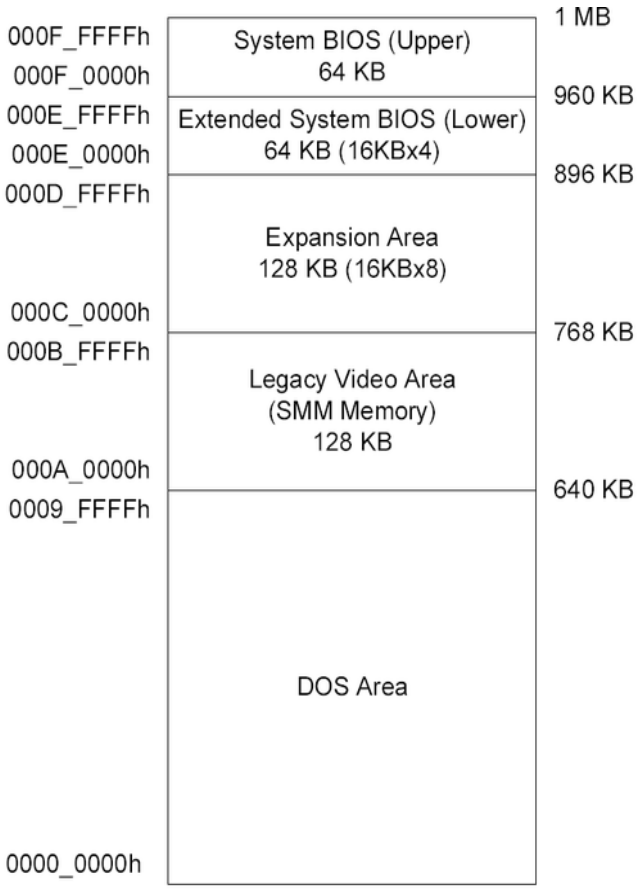

Рисунок 8: Унаследованный диапазон адресов

В современных компьютерах эта область выполняет две функции. Во-первых, она используется для совместимости со старым программным обеспечением и позволяет обращаться к видеопамяти.

Во-вторых, эта область может использоваться кодом режима управления системой (SMM). Когда процессор обращается к этим адресам в режиме SMM, GMCH передаёт его системной памяти, а не графическому контроллеру.

Следующие 128 Кбайт (адреса 000C_0000 — 000D_FFFF) — это ***область расширения*** (Expansion Area). По этим адресам могут располагаться ПЗУ BIOS внешних устройств. Любое обращение по этим адресам со стороны устройств, находящихся на PCI Express или DMI, всегда передаётся системной памяти.

Область расширенной системной BIOS (Extended System BIOS Area) занимает следующие 64 Кбайта (адреса 000E_0000 — 000E_FFFF). Эта область разделена на четыре сегмента по 16 Кбайт; обращения процессора к каждому сегменту могут направляться либо в оперативную память, либо на шину DMI и далее через ICH к ПЗУ BIOS, причём доступы на чтение и запись настраиваются отдельно.

Область системной BIOS (System BIOS Area) занимает старшие 64 Кбайта унаследованного диапазона (адреса 000F_0000 — 000F_FFFF). Она не делится на сегменты; доступы к этой области направляются либо в оперативную память, либо

на шину DMI и далее через ICH к ПЗУ BIOS, причём чтение и запись опять-таки настраиваются индивидуально.

Нижний диапазон адресов главной памяти.

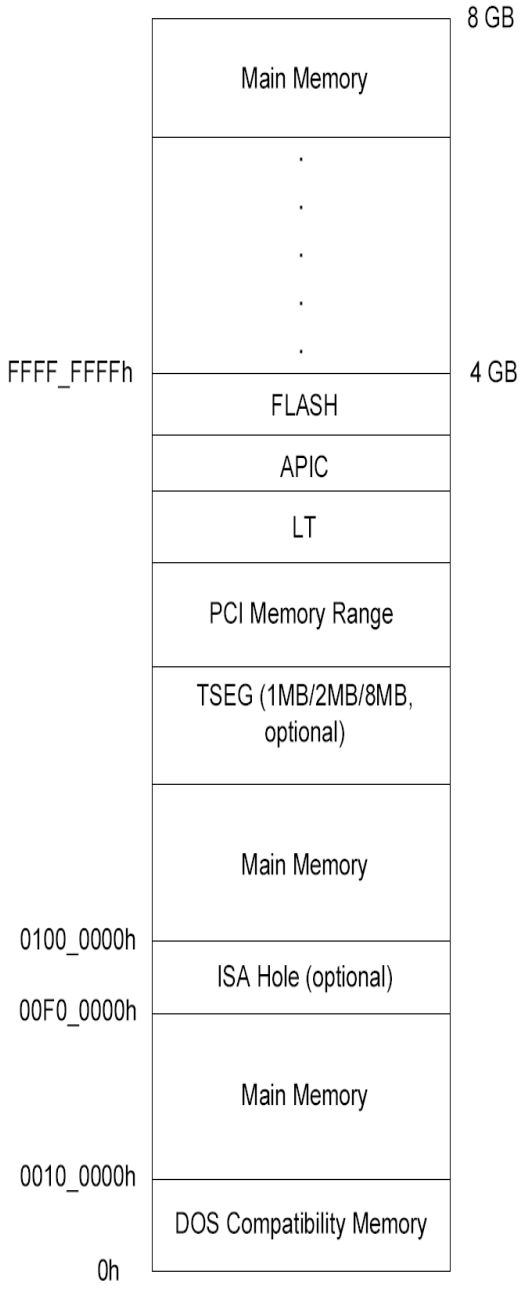

Рисунок 9: Карта нижнего диапазона адресов главной памяти

Нижняя часть **диапазона адресов главной памяти** (Main Memory Address Range) располагается от границы 1 Мбайта (начиная с адреса 0010_0000) до границы нижней доступной физической памяти (Top of Low Usable Physical Memory её устанавливает BIOS). Весь этот диапазон может быть прямо отображён на физическую память, однако здесь возможны исключения.

***Дыра ISA* (ISA Hole)** располагается между 15 и 16 Мбайтами адресного пространства памяти (00F0_0000 — 00FF_FFFF). Исторически её появление связано с микропроцессором 80286, способным в защищённом режиме адресовать до 16 Мбайт памяти, старший мегабайт которой был отведён под ПЗУ BIOS и некоторые другие нужды (в первую очередь под

видеопамять). Современные компьютеры в ней не нуждаются, поэтому обычно эта дыра отключена, а её адреса отображаются на физическое ОЗУ.

Необязательная область *TSEG* предназначена для использования кодом режима управления системой. Её размер может изменяться (для G45 он может составлять 1, 2 или 8 Мбайт), а располагается она в верхней части нижнего диапазона адресов главной памяти: её концом служит либо адрес, находящийся в TOLUD, либо адрес, с которого начинается память, отведённая для нужд встроенного графического контроллера (о ней — чуть ниже). Доступ к памяти в области TSEG возможен только со стороны процессора и только при нахождении его в режиме SMM.

Область памяти для нужд IGD выделяется только в том случае, если используется встроенный графический контроллер; на рисунке она не показана. Иногда эта область состоит из двух смежных участков. Первый предназначен для хранения таблицы переадресации графического контроллера и носит название «Pre-allocated Graphics GTT Stolen Memory», а его объём составляет 1 или 2 Мбайта. Второй участок, «Pre-allocated Graphics VGA Memory», имеет размер от 1 до 256 Мбайт, находится ниже первого участка и выше области TSEG и используется для хранения информации, необходимой для работы встроенного графического контроллера в режиме совместимости с VGA.

Нижний диапазон адресов памяти PCI

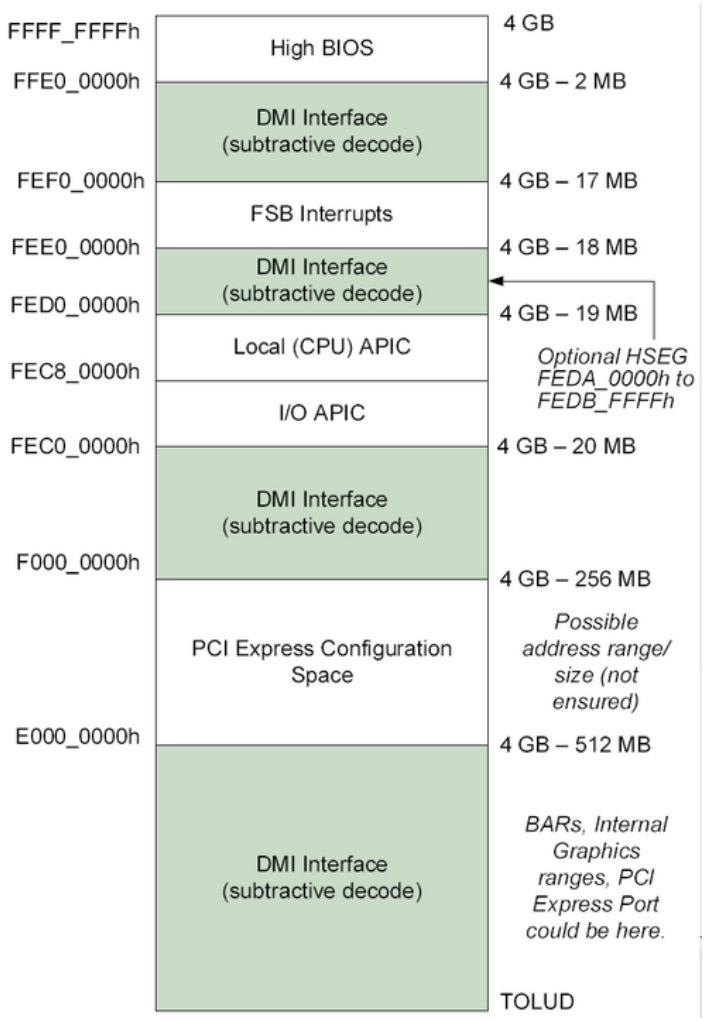

Рисунок 10: Карта нижнего диапазона адресов памяти PCI

Нижний **диапазон адресов памяти PCI** (PCI Memory Address Range) находится между TOLUD и границей 4 Гбайт (заканчивается адресом FFFF_FFFF). Большая его часть

отображается на шину DMI и служит для доступа к контроллерам, входящим в состав ICH или подключенным к шинам, обслуживаемым ICH, а также к ПЗУ BIOS. Однако ряд областей этого диапазона на DMI не отображается. Ниже различные участки адресов этого диапазона обсуждаются подробнее.

В самой верхней части этого диапазона, вплотную к границе 4 Гбайт, находится область ***верхней BIOS*** (High BIOS), под которую отведено 2 Мбайта (адреса FFE0_0000 — FFFF_FFFF). Эта область отображается на шину DMI и через ICH обеспечивает доступ к ПЗУ системной BIOS и BIOS контроллеров, если таковые имеются.

Один мегабайт в диапазоне адресов от FEE0_0000 до FEEF_FFFF отведён под ***пространство памяти прерываний*** *FSB* (FSB Interrupt Memory Space). Когда какое-либо устройство осуществляет запись по этим адресам, GMCH передаёт эту операцию на шину FSB как сообщение о прерывании.

Ещё один мегабайт (адреса FEC0_0000 — FECF_FFFF) зарезервирован для регистров ***конфигурационного пространства APIC*** (APIC Configuration Space). Это пространство разделено на две равные части. Нижняя половина (FEC0_0000 — FEC7_FFFF) служит для взаимодействия с IOAPIC, который физически обычно является частью ICH, хотя может быть выполнен и в виде отдельного контроллера. В любом случае обращения процессора к памяти по этим адресам передаются на шину DMI.

Верхняя половина этого мегабайта (FEC8_0000 — FECF_FFFF) может быть отображена на подключенный к нему «графический» порт PCI Express. Эта возможность используется, если там находится дополнительный IOAPIC. Однако на обычных ПК дополнительный IOAPIC не применяется, и запросы к этим адресам тоже передаются на шину DMI; впрочем, и на этой шине им обычно никакое устройство не соответствует. На практике можно считать, что эта область является зарезервированной.

Необязательная область *HSEG*, называемая иногда *верхним пространством памяти SMM* (High SMM Memory Space), может располагаться в диапазоне адресов FEDA_0000 — FEDB_FFFF. Если она имеется, то доступы к ней со стороны процессора, находящегося в режиме управления системой, отображаются на область ОЗУ с адресами 000A_0000 — 000B_FFFF (т.е. на нижнюю память режима управления системой, занимающую те же адреса, что унаследованная видеопамять, но доступные только в режиме SMM). Доступы к HSEG в других режимах являются недопустимыми. Если же область HSEG отсутствует, все обращения к этому диапазону адресов передаются, как и большинство других обращений в нижнем диапазоне адресов PCI, на шину DMI.

В оставшихся адресах может находиться ***конфигурационное пространство PCI Express*** (PCI Express Configuration Space). Его точный размер и местоположение зависят от настроек; на рисунке оно занимает адреса от E000_0000 до EFFF_FFFF. Подробнее о назначении и функциях этого конфигурационного пространства говорится в описании шины PCI Express.

Всё остальное пространство нижнего диапазона адресов памяти PCI отведено под память и регистры внешних устройств. Большинство из них либо являются частью ICH, либо находятся на шинах PCI и PCI Express, подключаемых к ICH, поэтому GMCH передаёт обращения по этим адресам на шину DMI. Однако часть устройств может находиться на шине PCI Express, подключенной к самому GMCH (обычно это дискретный видеоконтроллер). У

Верхний диапазон адресов главной памяти

Верхний **диапазон адресов главной памяти** (Main Memory Address Range) начинается с границы 4 Гбайт (адрес 1_0000_0000) и простирается вверх почти до конца физически установленной памяти (её адрес в процессе инициализации заносится BIOS в регистр TOM — Top Of Memory). До самого конца физической памяти этот диапазон не доходит по той причине, что какое-то количество самых верхних адресов используется для нужд одного из компонентов — так называемого Measurement Engine (ME, «измерительный

движок»). Объём этого изымаемого участка может доходить до 64 Мбайт, однако верхняя граница диапазона адресов главной памяти всегда выравнивается на границу 64 Мбайта, поэтому, если объём изымаемого участка меньше этой величины, неиспользованная память просто пропадает.

Диапазон адресов перемещённой главной памяти.

Нижний диапазон адресов памяти PCI «съедает» значительное количество адресного пространства памяти, лежащего ниже границы 4 Гбайт (на практике потеря может достигать 1,5 и даже 2 Гбайт в зависимости от типов и количества установленных контроллеров). Чтобы эта оперативная память не пропадала впустую, контроллер может обеспечить к ней доступ, используя адреса, лежащие сразу за верхним диапазоном адресов главной памяти: доступы к ним переотображаются на адреса физической памяти, лежащие ниже границы 4 Гбайта. Эта «спасённая» область ОЗУ называется **диапазоном адресов перемещённой главной памяти** (Main Memory Reclaim Address Range). С точки зрения операционной системы разницы между этой областью и верхним диапазоном адресов главной памяти нет, поэтому можно считать, что это единая область, начинающаяся с адреса 1_0000_0000 и заканчивающаяся адресом, находящимся в регистре TOLUD (Top of Upper Usable Physical Memory, вершина верхней пригодной для использования физической памяти).

Верхний диапазон адресов памяти PCI.

Верхний **диапазон адресов памяти PCI** начинается сразу за концом доступной физической памяти (TOUUD) и продолжается до конца адресуемого пространства памяти, т.е. до границы 64 Гбайт (адрес F_FFFF_FFFF). Эти адреса могут отображаться на шины PCI Express и DMI и использоваться для доступа к регистрам и памяти различных контроллеров, поддерживающих адресацию свыше 4 Гбайт.

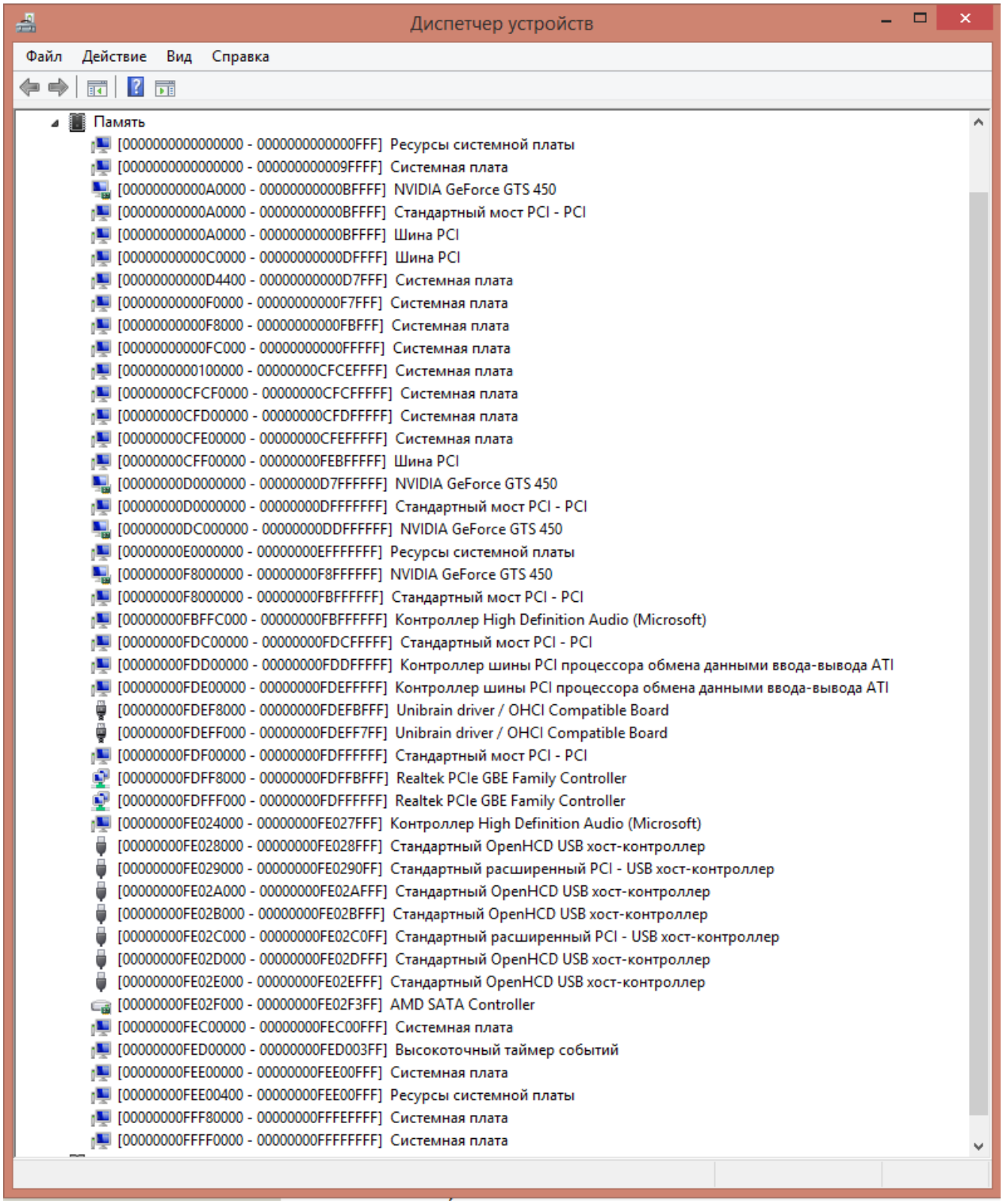

Рисунок 11: Отображение устройств в память. (OS Windows 8.1)

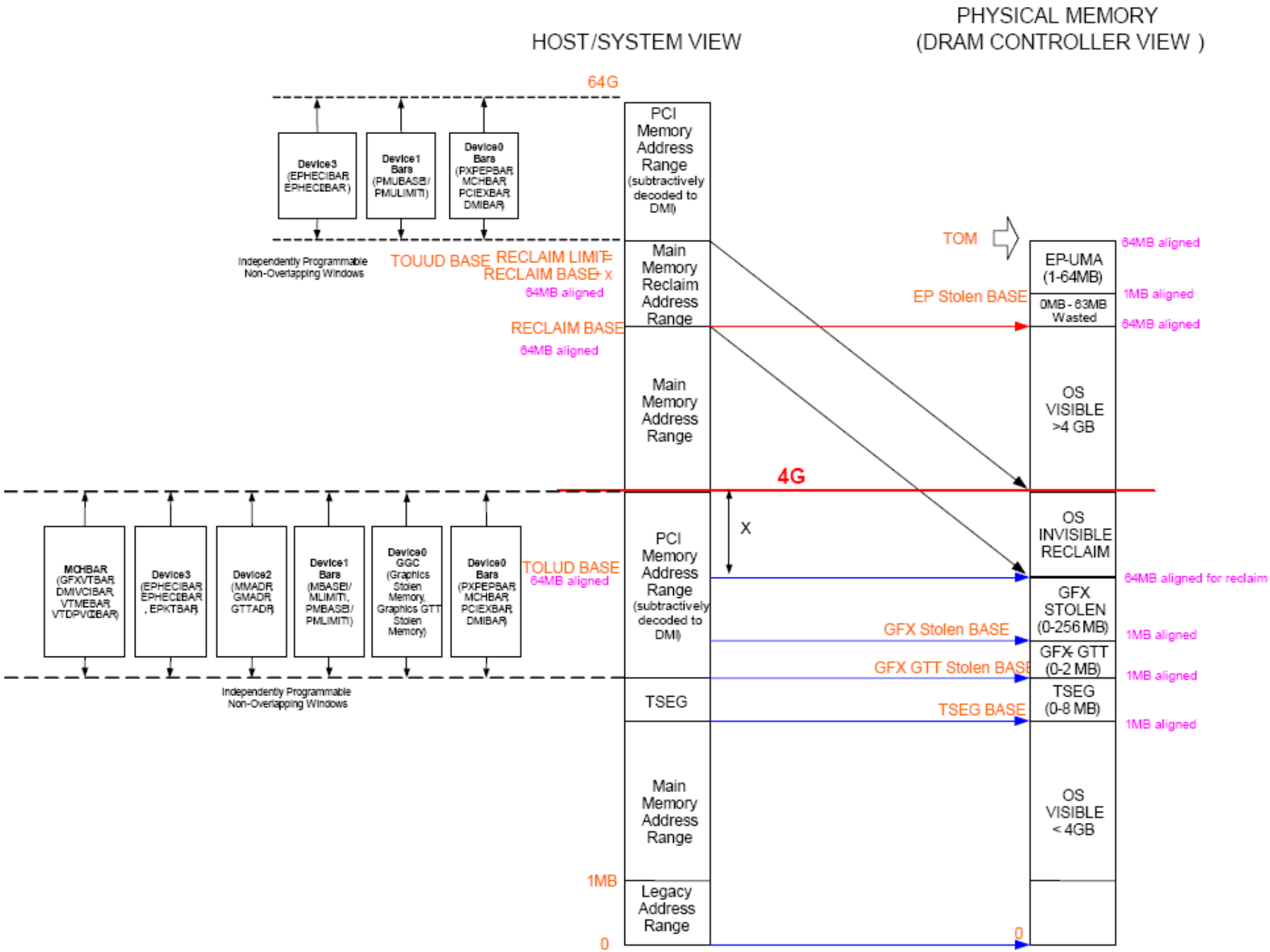

Рисунок 12: Типичная карта распределения физического адресного пространства памяти для GMCH G45

Вопросы для самоконтроля

1. Что такое основная память?
2. Что такое вторичная память?
3. Опишите иерархическую структуру памяти.
4. Что такое линейная логическая адресация памяти?
5. Что такое сегментация?
6. Зачем нужно связывание адресов?
7. На каких этапах может выполнятся привязка данных?
8. Какие функции должна выполнять система управления памятью?
9. Опишите схему с фиксированными разделами.
10. Что такое оверлейная структура?
11. Что такое свопинг?
12. Опишите работу страничной памяти.
13. Как формируется адрес при страничной адресации?
14. Как формируется адрес при сегментной адресации?
15. Как формируется адрес при сегментно-страничной адресации?
16. Для чего используется отображение памяти?
17. На какие участки делится память в системах с архитектурой IA-32&
18. Для чего используется унаследованный диапазон адресов?

Источники:

Микропроцессорные системы: Учебник / В.В. Гуров. - М.: НИЦ ИНФРА-М, 2016. - 336 с.: 60x90 1/16. - (Высшее образование: Бакалавриат) (Переплёт) ISBN 978-5-16-009950-7[1]

Лекция 8: Организация памяти компьютера. Простейшие схемы управления памятью: http://www.intuit.ru/studies/courses/2192/31/lecture/982?page=1

Карта распределения памяти: http://ru.osdev.wikia.com/wiki/Карта_распределения_памяти

1 Доступно в ЭБС Знаниум